

# 单调队列(广告牌问题模型)

题意：给你两个矩阵让你求出最大相同的子矩阵的面积。两个矩阵中的元素是两个1到n\*m的排列

思路：广告牌问题 先预处理出每个元素向上最远到达的地方。然后枚举底边，对每个底边遍历右边界，维护一个单调递增的单调队列，中间在维护一下每个元素最左到达的地方。每次出队的时候更新一下答案，出队的时候因为是新值小于队尾的值，所以是以自己为基准，所以更新答案是：

$$ans = \max(ans, (j - q[t].se) * q[t].fi);$$

最后在更新一下队列剩余元素的值即可，由于是单调递增的，所以队头最小，即以队头为准

$$ans = \max(ans, (j - q[h].se) * q[h].fi);$$

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
const int N = 1e6 + 10;
#define fi first
#define se second
#define pb push_back
int n,m;
int a[1111][1111],b[1111][1111];
struct uzi{
    int x,y;
}pA[N];
int f[N];
pair<int,int> q[1003];
int h,t;
int L[1003],ans=1;
int main() {
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++){for(int j=1;j<=m;j++)scanf("%d",&a[i][j]),pA[a[i][j]]={i,j};}
    for(int i=1;i<=n;i++)for(int j=1;j<=m;j++)scanf("%d",&b[i][j]);
    for(int i=1;i<=n;i++){
        for(int j=1;j<=m;j++){
            f[b[i][j]]=1;
            int le = b[i-1][j],dx=pA[b[i][j]].x,dy=pA[b[i][j]].y;
            if(a[dx-1][dy]==le)f[b[i][j]]=f[le]+1;
            ans=max(ans,f[b[i][j]]);
        }
    }
    for(int i=1;i<=n;i++){
        h=1;t=0;int can=1;
        for(int j=1;j<=m;j++){
            if(j==1)q[++t]={f[b[i][j]],1};
            else{
```

```

    int le = b[i][j-1];
    int dx=pA[b[i][j]].x,dy=pA[b[i][j]].y;
    if(a[dx][dy-1]!=le){
        while(h<=t){
            ans=max(ans,(j-q[h].se)*q[h].fi);
            h++;
        }
        h=1;t=0;
        q[++t]={f[b[i][j]],j};
        continue;
    }
    int pos=j;
    while(h<=t&&q[t].fi>=f[b[i][j]]){
        ans=max(ans,(j-q[t].se)*q[t].fi);
        t--;
        pos=q[t+1].se;
    }
    q[++t]={f[b[i][j]],pos};
}
}
while(h<=t){
    ans=max(ans,(m-q[h].se+1)*q[h].fi);
    h++;
}
}
printf("%d\n",ans);
return 0;
}

```